

Санкт-Петербургский государственный университет
Факультет Прикладной математики – процессов управления
**Кафедра теории систем управления
электрофизической аппаратурой**

Макарова Анастасия Сергеевна

Выпускная квалификационная работа бакалавра

**Метод оптического потока в задаче
построения поля скоростей**

Направление 010900

«Прикладные математика и физика»

Руководитель научной программы,
доктор физ.-мат. наук,
профессор

Егоров Н. В.

Научный руководитель,
доктор физ.-мат. наук,
профессор

Овсянников Д. А.

Рецензент,
доктор физ.-мат. наук,
профессор

Котина Е. Д.

Санкт-Петербург

2016

Содержание

Введение	3
Постановка задачи	5
Обзор литературы	7
1 Дифференциальный метод оптического потока	8
1.1 Схема решения задачи	8
1.2 Расчет разностной схемы	9
1.3 Решение системы линейных алгебраических уравнений . . .	12
1.4 Сходимость метода последовательной верхней релаксации .	15
2 Реализация метода и проверка результатов	17
2.1 Алгоритм программы	17
2.2 Пирамиды изображений	17
2.3 Подсчет смешанных производных и производных второго по- рядка	19
2.4 Расчет поля скоростей	19
2.5 Пополнение значений оптического потока	20
2.6 Тестовые задачи	21
2.7 Построение входных изображений	24
2.8 Полученные результаты на тестовых задачах	27
Заключение	32
Литература	33

Введение

С быстрым ростом развития информационных технологий появилась необходимость в автоматическом и мгновенном определении, какие действия происходят на последовательности кадров. Таким образом, с помощью данной последовательности необходимо воссоздать запечатленное на них пространство и изменения, происходящие с ним с течением времени. С данной задачей и связано появление такого понятия как оптический поток.

На сегодняшний день это словосочетание применимо во многих областях технической разработки, и в большом количестве источников, предоставляющих информацию об оптических потоках, дается свое определение данному понятию. Для полного понимания необходимо привести самое общее определение: *оптический поток* — это визуальное отображение видимого движения объектов или поверхностей, которое возникает при перемещении объектов или наблюдателя относительно друг друга.

Таким образом, оптический поток может дать важную информацию о пространственных положениях объектов, о характере его движения, о форме рассматриваемого объекта, а также о структуре сцены. Для оптических потоков характерно расширение границ их применений: они используются для создания видео- и фото-спецэффектов, в областях сжатия видео и анализа движения, в компьютерном зрении, в компьютерной томографии и многих других областях. Конечно, большое многообразие алгоритмов, используемых для их получения, является косвенным подтверждением использования оптических потоков в различных областях с распознаванием образов и сигналов.

Оптический поток определяется как «поток» уровней яркости в плоскости изображений. Заметим, что последовательность изображений, упорядоченных во времени, дает возможность оценить зарегистрированное движение или как мгновенную скорость, или как численное смещение.

Градиент изображения является одним из фундаментальных понятий в обработке изображений. Поэтому, в данной работе реализована и построена математическая модель для определения поля скоростей на постоян-

стве градиента интенсивности, то есть скорость изменения интенсивности постоянна.

Тема данной выпускной работы связана с построением поля скоростей с помощью оптических потоков. Решение полученных уравнений в ходе имеющегося метода, если применить к ним теорию разностных схем, сводится к решению системы линейных уравнений. Линейные системы решаются блочными итерационными методами, и далее показывается сходимость этих методов к единственному решению системы.

Основной целью данной работы является построение поля скоростей для входных изображений с помощью дифференциального метода оптического потока. Существенной целью является реализация данного метода в среде разработки Microsoft Visual Studio на языке C++ с помощью библиотеки OpenCV для проверки метода и анализа полученных результатов.

Работа состоит из двух глав, приложения и списка литературы.

В первой главе приведены главные теоретические положения и основные принципы отыскания оптического потока. Модель, рассматриваемая в рамках представленной работы, расширяет возможности исследования потоков, где плотность интенсивности вдоль траекторий меняется.

Во второй главе рассматриваются общие вопросы компьютерной реализации выбранного метода для решения поставленной задачи. Также приведены результаты обработки с помощью реализованной программы ряда тестовых задач, представляющие собой изображения с различными формами движения.

В заключении подводятся итоги исследования, проведенного в ходе выполнения данной работы, а также формируются окончательные выводы.

Постановка задачи

Пусть с течением времени происходят изменения положения объекта, характеризующиеся смещениями на малые приращения. Таким образом, рассмотрим математическую модель в виде системы дифференциальных уравнений:

$$\begin{cases} \dot{x} = u(t, x, y), \\ \dot{y} = v(t, x, y), \end{cases} \quad (1)$$

где x и y — пространственные координаты, t — время захвата кадра в видеопотоке, а u и v являются искомыми функциями, определяющими поле скоростей.

Также сделаем допущение, что градиент плотности распределения интенсивности вдоль траекторий системы (1) остается постоянным:

$$\left. \frac{d}{dt} \begin{pmatrix} \frac{\partial \rho(t, x, y)}{\partial x} \\ \frac{\partial \rho(t, x, y)}{\partial y} \end{pmatrix} \right|_{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (2)$$

где под функцией $\rho(t, x, y)$ понимается яркость объекта в точке (x, y) в момент времени t , а индекс 1 при вертикальной черте формулы (2) означает, что производная берется в силу системы (1). Тогда распишем полные производные компонент вектора градиента:

$$\begin{cases} \frac{d}{dt} \left(\frac{\partial \rho(x, y, t)}{\partial x} \right) = \frac{\partial^2 \rho(x, y, t)}{\partial x^2} u + \frac{\partial^2 \rho(x, y, t)}{\partial x \partial y} v + \frac{\partial^2 \rho(x, y, t)}{\partial x \partial t}; \\ \frac{d}{dt} \left(\frac{\partial \rho(x, y, t)}{\partial y} \right) = \frac{\partial^2 \rho(x, y, t)}{\partial y^2} v + \frac{\partial^2 \rho(x, y, t)}{\partial y \partial x} u + \frac{\partial^2 \rho(x, y, t)}{\partial y \partial t}. \end{cases} \quad (3)$$

Каждая компонента этого вектора равна нулю. В результате получается, что приведенное выше предположение имеет математическую формулировку вида:

$$\begin{cases} \rho_{xx}u + \rho_{xy}v = -\rho_{xt}, \\ \rho_{yx}u + \rho_{yy}v = -\rho_{yt}, \end{cases} \quad (4)$$

где $\rho_{xx}, \rho_{xt}, \rho_{xy}, \rho_{yy}, \rho_{yt}$ — обозначения для частных производных второго порядка плотности оптического потока ρ . Полученная система накладывает ограничения на компоненты поля скоростей, иными словами мы получили систему уравнений оптического потока.

Исходя из вышеописанной модели, необходимо рассмотреть обратную задачу определения поля скоростей системы (1), то есть найдем функции u, v по известной функции интенсивности $\rho(t, x, y)$. Для этого будем использовать метод регуляризации, описанный А. Н. Тихоновым [9]. Для этого рассмотрим интегральный функционал:

$$J(u, v) = \iint_M (\phi^2 + \alpha^2 \psi^2) dx dy, \quad (5)$$

где M — некоторая область из R^2 в момент времени t , а $\phi^2 = (\rho_{xt} + \rho_{xx}u + \rho_{xy}v)^2 + (\rho_{yt} + \rho_{yx}u + \rho_{yy}v)^2$ — интегральная невязка уравнений оптического потока. В полученном функционале мерой гладкости будет функция ψ , имеющая вид:

$$\psi^2 = u_x^2 + u_y^2 + v_x^2 + v_y^2, \quad (6)$$

где α^2 — весовой параметр, определяющий уровень значимости сглаживающей части функционала.

Таким образом, предстоит решить задачу минимизации функционала (5).

Обзор литературы

Построение поля скоростей на основе определения оптического потока рассматривается во многих работах. Общую постановку и ход решения указанной задачи градиентным методом оценки оптического потока можно найти в работе [1].

В работе [2] J.L. Barron, D.J. Fleet и S.S. Beauchemin провели анализ методов вычисления оптических потоков с точки зрения векторного поля, а также с точки зрения точности приведенных методов.

Дифференциальные локальные методы оптического потока делятся на несколько видов:

- метод оптического потока (Horn–Schunck) и модифицированные методы, основанные на нем, базируются на минимизации функционала с различными ограничениями на данные и на гладкость получаемого векторного поля [3][4];
- алгоритм Лукаса—Канаде основан на предположении о том, что в определенной окрестности пикселя оптический поток одинаков, и решается методом наименьших квадратов [1];
- метод Buxton–Buxton строится на выделении границ объектов и на анализе движений их в последовательности изображений [5].

Монография [6] посвящена моделированию динамических процессов в радионуклидных исследованиях. На основе этой модели рассматриваются блочные итерационные методы для решения системы, к которой сводится задача решения уравнений оптического потока, то есть уравнений в частных производных, и их сходимость.

В статье, лежащей в основе данной работы, [7] Котиной Е.Д. и Пасечной Г.А. рассмотрены два подхода к определению поля скоростей, а именно: первый подход, основанный на предположении о постоянстве плотности распределения радиофармпрепарата вдоль траекторий движения, и второй — на предположении о постоянстве ее градиента.

1 Дифференциальный метод оптического потока

1.1 Схема решения задачи

Интегральный функционал удобно представить в следующем виде:

$$J(u, v) = \iint_M (\phi^2 + \alpha^2 \psi^2) dx dy = \iint_M F(u, v, u_x, u_y, v_x, v_y) dx dy. \quad (7)$$

Так как функции, участвующие в формуле (7), равны $\phi^2 = (\rho_{xt} + \rho_{xx}u + \rho_{xy}v)^2 + (\rho_{yt} + \rho_{yx}u + \rho_{yy}v)^2$ и $\psi^2 = u_x^2 + u_y^2 + v_x^2 + v_y^2$, то подынтегральное выражение имеет вид:

$$F(u, v, u_x, u_y, v_x, v_y) = (\rho_{xt} + \rho_{xx}u + \rho_{xy}v)^2 + (\rho_{yt} + \rho_{yx}u + \rho_{yy}v)^2 + \alpha^2(u_x^2 + u_y^2 + v_x^2 + v_y^2). \quad (8)$$

Минимизация интеграла данного вида является задачей вариационного исчисления. Соответствующие уравнения Эйлера имеют вид:

$$\begin{cases} F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0, \\ F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} = 0. \end{cases} \quad (9)$$

Запишем уравнения Эйлера с учетом вида подынтегральной функции (8):

$$\begin{aligned} -\alpha^2 \Delta u + (\rho_{xx}^2 + \rho_{yx}^2)u + \rho_{xy}(\rho_{xx} + \rho_{yy})v &= -\rho_{xx}\rho_{xt} - \rho_{yx}\rho_{yt}, \\ -\alpha^2 \Delta v + (\rho_{xy}^2 + \rho_{yy}^2)v + \rho_{xy}(\rho_{xx} + \rho_{yy})u &= -\rho_{xy}\rho_{xt} - \rho_{yy}\rho_{yt}, \end{aligned} \quad (10)$$

где Δ — оператор Лапласа, определяется как $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. Иными словами, теперь задача нахождения поля скоростей сводится к решению системы (10). Эту пару эллиптических уравнений второго порядка можно решить итеративными методами.

Для того чтобы задача была корректно определена, необходима единственность существующего решения. Уравнения такого вида обычно имеют

бесконечное множество решений, пока отсутствуют граничные условия. Таким образом, для корректности задачи в этом случае необходимо наложить граничные условия на систему.

Вследствие того, что здесь имеет место система эллиптических уравнений второго порядка, можно задать значения функции на простой замкнутой границе интересующей нас области, то есть:

$$\begin{aligned} u(x, y) &= f(x, y), \quad (x, y) \in \gamma; \\ v(x, y) &= g(x, y), \quad (x, y) \in \gamma. \end{aligned} \tag{11}$$

Здесь γ — граница области M , а функции $f(x, y)$ и $g(x, y)$ — наперед заданные функции.

В свою очередь, другой возможностью создания корректной задачи является также задание значений нормальной производной функций u и v , которая представлена в монографии [10].

1.2 Расчет разностной схемы

Рассмотрим систему (10), где $(x, y) \in M$, а функции u и v заданы на границе области M , которую обозначим как γ , и принимают нулевые значения на этой границе. Обратим внимание на то, что в таком случае функции $f(x, y)$ и $g(x, y)$ в системе (11) равны нулю. Правая часть уравнений (10) представляет собой известную функцию от x , y . Учитывая дискретность входных данных, необходимо аппроксимировать оператор Лапласа конечно-разностной схемой. Решение задачи осуществляется с использованием схемы второго порядка аппроксимации на пятиточечном шаблоне на равномерных сетках по x и y . Есть равномерная сетка с узлами $x_i = ih$, $y_j = jh$, $i, j = 1, \dots, n$, где h — шаг сетки, равный единичному изменению расстояния. Под единичным изменением будет пониматься приращение вдоль какой-либо оси величиной в один пиксель изображения, т.е. $h = 1$. Если обозначить через $u(i, j)$, $v(i, j)$ приближение к решению задачи в точке (i, j) , принадлежащей сетке, то частные производные имеют вид (рис. 1):

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) \approx u(i+1, j) - 2u(i, j) + u(i-1, j);$$

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) \approx u(i, j+1) - 2u(i, j) + u(i, j-1).$$

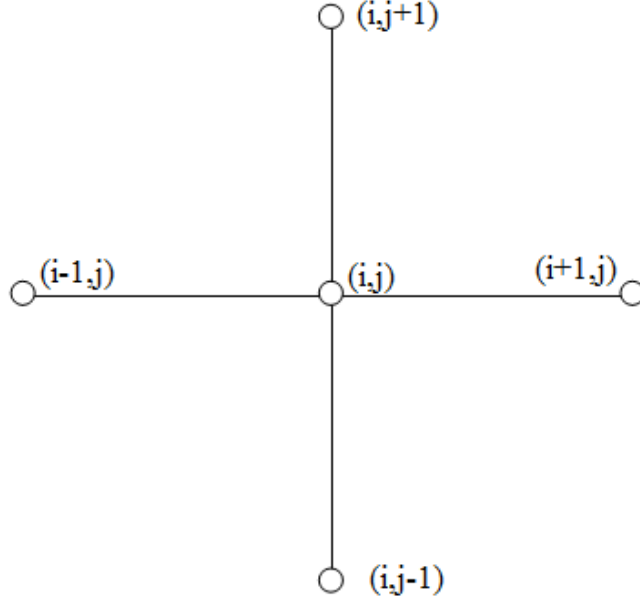


Рис. 1. Аппроксимация вторых частных производных u и v с помощью компонент оптического поля в соседних точках.

Следующим шагом для расчета разностной схемы является построение сужения функций ρ_{xx} , ρ_{yx} , ρ_{xt} , ρ_{yy} , ρ_{yt} , $u(x, y)$, $v(x, y)$ на упомянутую ранее равномерную сетку. Здесь будем обозначать функцию интенсивности, находящуюся на пересечении i -й строки, j -го столбца и k -го момента времени за $\rho_k(i, j)$, $i, j = 1, \dots, n$. Таким образом, заменяя оператор Лапласа конечными разностями и сужениями функций, получаем переход от уравнений в частных производных к системе линейных уравнений вида:

$$\left\{ \begin{array}{l} -\alpha^2(u(i-1, j) + u(i+1, j) + u(i, j-1) + u(i, j+1)) + \\ + (4\alpha^2 + \rho_{xx}^2(i, j) + \rho_{yx}^2(i, j))u(i, j) + (\rho_{xx}(i, j) + \rho_{yy}(i, j)) \times \\ \times \rho_{xy}(i, j)v(i, j) = -\rho_{xx}(i, j)\rho_{xt}(i, j) - \rho_{yx}(i, j)\rho_{yt}(i, j); \\ -\alpha^2(v(i-1, j) + v(i+1, j) + v(i, j-1) + v(i, j+1)) + \\ + (4\alpha^2 + \rho_{xy}^2(i, j) + \rho_{yy}^2(i, j))v(i, j) + (\rho_{xx}(i, j) + \rho_{yy}(i, j)) \times \\ \times \rho_{xy}(i, j)u(i, j) = -\rho_{xy}(i, j)\rho_{xt}(i, j) - \rho_{yy}(i, j)\rho_{yt}(i, j); \end{array} \right. \quad (12)$$

$$i, j = 2, \dots, n - 1.$$

Эта система имеет размерность $2(n - 2) \times 2(n - 2)$ с количеством неизвестных $2n^2$. Необходимо замкнуть систему. С этой целью добавим граничные условия вида (11) с нулевыми значениями:

$$\begin{aligned} i = 1, \quad i = n : u(1, j) = 0, \quad u(n, j) = 0, \\ v(1, j) = 0, \quad v(n, j) = 0, \quad j = 2, \dots, n - 1, \\ j = 1, \quad j = n : u(i, 1) = 0, \quad u(i, n) = 0, \\ v(i, 1) = 0, \quad v(i, n) = 0, \quad i = 2, \dots, n - 1. \end{aligned} \quad (13)$$

При этом, значения $u(1, n) = 0$, $u(n, 1) = 0$, $v(1, n) = 0$, $v(n, 1) = 0$ при подсчете не учитываются, и их инициализация может происходить в любой момент, не повлияв на результаты.

Для последовательности изображений, состоящей из двух кадров, частные производные второго порядка считаются таким образом:

$$\begin{aligned} \rho_{xx}(i, j) &\approx \rho_1(i - 1, j) - 2\rho_1(i, j) + \rho_1(i + 1, j), \\ \rho_{yy}(i, j) &\approx \rho_1(i, j - 1) - 2\rho_1(i, j) + \rho_1(i, j + 1). \end{aligned} \quad (14)$$

Смешанные производные в данной задаче представляют собой выражения вида:

$$\begin{aligned} \rho_{xy} &\approx \frac{1}{4}(\rho_1(i + 1, j + 1) + \rho_1(i - 1, j - 1) - \rho_1(i + 1, j - 1) - \rho_1(i - 1, j + 1)); \\ \rho_{xt} &\approx \frac{1}{4}(\rho_2(i + 1, j) + \rho_1(i - 1, j) - \rho_2(i - 1, j) - \rho_1(i + 1, j)); \\ \rho_{yt} &\approx \frac{1}{4}(\rho_2(i, j + 1) + \rho_1(i, j - 1) - \rho_2(i, j - 1) - \rho_1(i, j + 1)). \end{aligned} \quad (15)$$

В конечном счете, можно вычислить все приближенные значения частных производных второго порядка ρ_{xx} , ρ_{xt} , ρ_{xy} , ρ_{yy} , ρ_{yt} , используя значения плотности в соседних точках сетки по выбранной схеме.

Согласно предположению о том, что функции u и v заданы на границе области, только $2(n - 1)^2$ переменных $u(i, j)$, $v(i, j)$, $i, j = 1, \dots, n$, во внутренних точках сетки являются неизвестными в уравнениях (12).

Таким образом, получается линейная система из $2n^2$ уравнений, решение которой дает приближение к решению системы (10) в точках сетки. В результате дискретизации системы дифференциальных уравнений в частных производных (1.2) получили линейную систему разностных уравнений (12). Далее рассматривается решение полученной системы.

1.3 Решение системы линейных алгебраических уравнений

Систему представленную выражением (12) можно записать в виде:

$$\begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} d \\ e \end{pmatrix}, \quad (16)$$

где

$$u = (u_1, \dots, u_{n^2})^T = (u_{11}, \dots, u_{1n}, u_{21}, \dots, u_{2n}, \dots, u_{n1}, \dots, u_{nn})^T,$$

$$v = (v_1, \dots, v_{n^2})^T = (v_{11}, \dots, v_{1n}, v_{21}, \dots, v_{2n}, \dots, v_{n1}, \dots, v_{nn})^T$$

— значения искомых функций; вектор-столбец в правой части системы имеет компоненты вида:

$$d = (d_1, \dots, d_{n^2})^T = (d_{11}, \dots, d_{1n}, d_{21}, \dots, d_{2n}, \dots, d_{n1}, \dots, d_{nn})^T,$$

$$e = (e_1, \dots, e_{n^2})^T = (e_{11}, \dots, e_{1n}, e_{21}, \dots, e_{2n}, \dots, e_{n1}, \dots, e_{nn})^T.$$

Здесь под элементами $d_{i,j}$ и $e_{i,j}$ понимаются:

$$d_{i,j} = -\rho_{xx}(i,j)\rho_{xt}(i,j) - \rho_{yx}(i,j)\rho_{yt}(i,j), \quad i, j = \overline{1, n}, \quad (17)$$

$$e_{i,j} = -\rho_{xy}(i,j)\rho_{xt}(i,j) - \rho_{yy}(i,j)\rho_{yt}(i,j), \quad i, j = \overline{1, n}, \quad (18)$$

Матрица B — диагональная, т.е.

$$\begin{aligned} b_{sr} &= 0, \quad s, r = 1, \dots, n^2, \\ b_{ss} &= \rho_{xy}(i,j)(\rho_{xx}(i,j) + \rho_{yy}(i,j)), \quad i, j = \overline{1, n}, \quad s = \overline{1, n^2}. \end{aligned} \quad (19)$$

Матрицы A, C — разреженные и отличаются диагональными элементами, то есть $a_{sr} = c_{sr}$ и ненулевые из них $a_{sr} = c_{sr} = -\alpha^2$. Диагональные элементы матрицы A :

$$a_{ss} = 4\alpha^2 + \rho_{xx}^2(i, j) + \rho_{yx}^2(i, j), \quad i, j = \overline{1, n}, \quad s = \overline{1, n^2}, \quad (20)$$

матрицы C :

$$c_{ss} = 4\alpha^2 + \rho_{yy}^2(i, j) + \rho_{yx}^2(i, j), \quad i, j = \overline{1, n}, \quad s = \overline{1, n^2}. \quad (21)$$

Для пояснения приведем вид матрицы системы (16) при $n = 3$, изображенный на рис. 2.

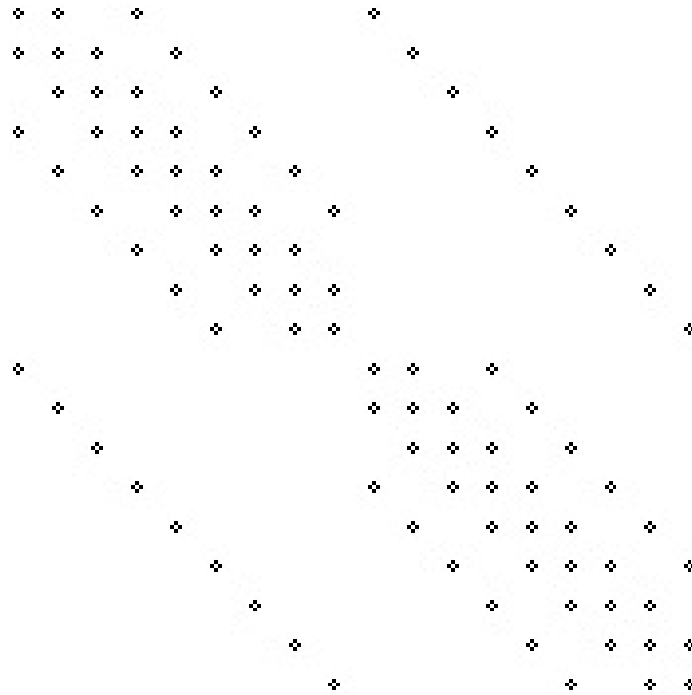


Рис. 2. Вид матрицы системы (16) размером 3×3 .

Для большего удобства можно привести представленную систему к новому виду, где за неизвестную берется вектор с двумя компонентами, имеющими значения поля скоростей, принадлежащего одной точке, по x и y координатам. Пусть

$$z = (z_1, \dots, z_{n^2})^T, \quad z_1 = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \dots, z_{n^2} = \begin{pmatrix} u_{n^2} \\ v_{n^2} \end{pmatrix},$$

$$q = (q_1, \dots, q_{n^2})^T, \quad q_1 = \begin{pmatrix} d_1 \\ e_1 \end{pmatrix}, \dots, q_{n^2} = \begin{pmatrix} d_{n^2} \\ e_{n^2} \end{pmatrix}.$$

Тогда система (16) примет вид:

$$Hz = q, \quad (22)$$

где компоненты матрицы H :

$$H_{ss} = \begin{pmatrix} a_{ss} & b_{ss} \\ b_{ss} & c_{ss} \end{pmatrix}, \quad s = \overline{1, n^2}, \quad (23)$$

$$H_{sr} = \begin{pmatrix} a_{sr} & b_{sr} \\ b_{sr} & c_{sr} \end{pmatrix}, \quad s \neq r, \quad s = \overline{1, n^2}, \quad r = \overline{1, n^2}. \quad (24)$$

Представим матрицу H следующим образом:

$$H = D - E - F, \quad (25)$$

где D — матрица, состоящая из диагональных блоков:

$$D = \begin{pmatrix} H_{11} & 0 & \dots & 0 \\ 0 & H_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & H_{n^2 n^2} \end{pmatrix},$$

$$E + F = \begin{pmatrix} 0 & H_{12} & \dots & H_{1n^2} \\ H_{21} & 0 & \dots & H_{2n^2} \\ \vdots & \vdots & \ddots & \vdots \\ H_{n^2 1} & \dots & H_{n^2, n^2-1} & 0 \end{pmatrix},$$

где

$$H_{ii} = \begin{pmatrix} a_{ii} & b_{ii} \\ b_{ii} & c_{ii} \end{pmatrix},$$

$$H_{ij} = \begin{pmatrix} a_{ij} & b_{ij} \\ b_{ij} & c_{ij} \end{pmatrix}.$$

Таким образом, систему (22) с матрицей вида (23),(24) представляется возможным решать более удобным методом — блочным итерационным методом последовательной верхней релаксации.

1.4 Сходимость метода последовательной верхней релаксации

Блочный метод последовательной верхней релаксации решения системы (22), отвечающий введенному разбиению, имеет вид:

$$H_{ss}z_s^{k+1} = \omega \left(- \sum_{r < s}^{n^2} H_{sr}z_r^{k+1} - \sum_{r > s}^{n^2} H_{sr}z_r^k + q_s \right) + (1 - \omega)H_{ss}z_s^k, \quad (26)$$

$$s = \overline{1, n^2}, k = 0, 1, 2, \dots, \omega \in (1, 2)$$

или в матричном представлении:

$$Dz^{k+1} = \omega(Ez^{k+1} + Fz^k + q) + (1 - \omega)Dz^k. \quad (27)$$

В дополнение, можно заметить что при $\omega = 1$ метод переходит в метод Гаусса-Зейделя. Также при $\omega \in (0, 1)$ он модифицируется в метод последовательной нижней релаксации.

Для его сходимости в случае, когда матрица B — диагональная, а матрицы A и C отличаются диагональными элементами, достаточно, чтобы выполнялись условия, приведенные в книге [6] и в статье [8]:

1. $a_{ss}c_{ss} - b_{ss}^2 > 0, a_{ss} > 0, c_{ss} > 0, s = 1, \dots, n^2,$
2. $\frac{a_{ss}+c_{ss}}{2} \geq \sum_{r=1, r \neq s}^{n^2} |a_{sr}| + \sqrt{\left(\frac{a_{ss}-c_{ss}}{2}\right)^2 + b_{ss}^2}, s = 1, \dots, n^2$ и для некоторого i выполняется это неравенство как строгое,
3. условие «блочной» неприводимости.

Если матрица H удовлетворяет всем вышеперечисленным условиям, тогда итерационный метод последовательной верхней релаксации сходится при любом начальном приближении z_0 к единственному решению системы (22).

Обратим внимание на то, что под блочной неприводимостью понимается неравенство нулю всех блоков матрицы, то есть $\|H_{ij}\| \neq 0$ для любых $i, j = 1, \dots, n^2$. Здесь возьмем за норму матрицы H_{ij} подчиненную евклидовой норму, в этом конкретном случае она равна $\|H_{ij}\| = |a_{ij}|$.

2 Реализация метода и проверка результатов

Содержание данной главы представляет собой разработку программы, реализующей данный алгоритм, на языке C++ в среде разработки Microsoft Visual Studio с использованием библиотеки OpenCV. Второй частью представленной главы является полный анализ результатов различными способами.

2.1 Алгоритм программы

Схема программы имеет вид:

1. Создание пирамиды изображений;
2. Подсчет смешанных производных и производных второго порядка;
3. Для каждого уровня пирамиды расчет поля скоростей, начиная с наибольшего;
4. С каждым расчетом пополнение значений оптического потока.

Далее рассмотрим каждый пункт схемы программы подробнее.

2.2 Пирамиды изображений

Для увеличения чувствительности предложенного алгоритма, а именно для правильного отыскания компонент вектора поля скоростей при смещениях, превышающих малое приращение, на изображениях большого разрешения строится пирамида изображений. Она является набором снимков в порядке уменьшения разрешения [12]. В основе данной пирамиды лежит исходное изображение со свойственным ему масштабом (рис. 3). Пусть основание имеет уровень J , то есть размер его равен $2^J \times 2^J$ или $N \times N$, где $J = \log_2 N$.

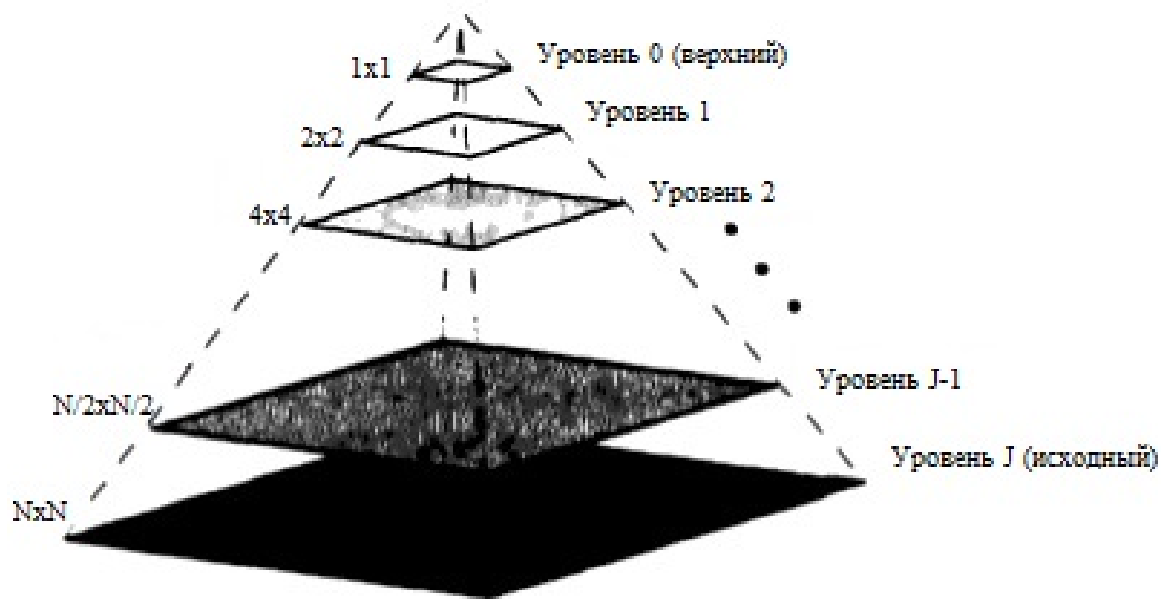


Рис. 3. Структура пирамиды изображений.

Алгоритм получения каждого уровня пирамиды состоит в том, что следующий уровень пирамиды меньшего разрешения образуется с помощью фильтрации предыдущего снимка и прореживания выборки со значением два. Под последним понимается отбрасывание каждого второго отсчета. Фильтрация производится различными методами: фильтром Гаусса; усреднением по окрестности; фильтром, увеличивающим четкость (увеличивает разницу значений на границах объектов); или есть возможность не использовать фильтрацию вовсе. В данной работе используется фильтр Гаусса с целью снижения уровня шума, но при этом он производит размытие входного изображения. Данная фильтрация производится с помощью вложенной функции

$$cvSmooth(img1, img2, CV_GAUSSIAN, 3, 3)$$

библиотеки OpenCV. Здесь входные данные имеют значения:

- *img1* — входное изображение, которое необходимо профильтровать;
- *img2* — это выходное изображение, в которое записывается результат проведенной работы;
- *CV_GAUSSIAN* — это метка, указывающая на то, что необходим Гауссовский фильтр;

- два последних входных параметра указывают на то, какими размерами берется блок для фильтрации. В представленном примере это блок 3 на 3 пикселя.

Заметим, что пирамиды изображений будут рассчитываться для каждого кадра из последовательности, представленной для нахождения поля скоростей. Теперь, используя представленную схему построения пирамиды, будем понимать под смещением пикселя результирующее значение вектора перемещения, полученного с помощью усреднения значений данного вектора по всем изображениям из пирамиды.

2.3 Подсчет смешанных производных и производных второго порядка

По формулам, представленным в предыдущей главе под номерами (14) и (15), считаются частные производные второго порядка и смешанные производные. Впоследствии для оптимального использования памяти, данные, рассчитанные по представленным формулам, сохраняются в текстовые файлы формата txt, и на следующих этапах работы программы с них считываются необходимые элементы данных.

Эта часть программы запускается для каждого уровня пирамиды изображений, то есть для каждого размера изображений создаются отдельные файлы с посчитанными значениями производных. Это сделано и с той целью, чтобы на уровне тестирования была возможность не просчитывать вновь производные для одних и тех же входных данных.

2.4 Расчет поля скоростей

На данном шаге нам важно значение параметра α , которое характеризует важность сглаживающей части функционала. Этот параметр должен быть большим при точном измерении яркости и малым при сильном зашумлении измерений. Так как для разных входных последовательностей α подбирается свой для лучшего отображения оптического потока, то он

является входным параметром для расчета поля скоростей. В текущей работе за параметр гладкости было взято значение равное 0.1.

Еще одним входным значением для функции, реализующей текущий этап работы, является количество итераций для просчета методом последовательной верхней релаксации. Вторым критерием остановки расчетов итеративным методом может быть заданная точность, с которой необходимо получить решение. Но так как нас интересует не очень высокая точность (можно ограничиться значением 0.1) и операция сравнения увеличивает работу программы на ощутимое время, то в нашем случае достаточно задавать количество многократных повторений расчетов представленным методом.

Дальнейшие действия, совершаемые представленной функцией, представляют собой заполнения матриц A , B , C и векторов d и e по формулам (17)-(21). Теперь по представленным матрицам можно инициализировать матрицу H , имеющую вид (23)-(24). В последующей работе программы матрицы A , B , C и вектора d и e не понадобятся, следовательно, для оптимальности использования памяти, освободим из-под них выделенный участок памяти.

С целью проверки заполненности матрицы H , была также написана функция для отображения ее структуры. На рис. 4 приведен результат работы вышеупомянутой функции.

Для подтверждения сходимости системы (22) производится проверка значений элементов матрицы H по всем условиям, представленным в последнем пункте первой главы.

Этот этап работы программы совершается для каждого уровня пирамиды.

2.5 Пополнение значений оптического потока

Вследствие того, что на каждом уровне пирамиды идет расчет поля скоростей, необходимо делать каждый раз корректировку данных, то есть

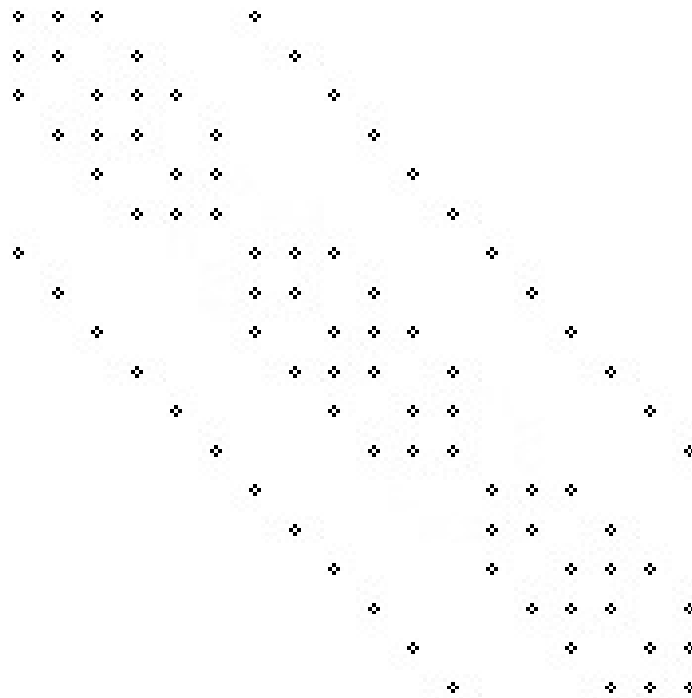


Рис. 4. Структура матрицы H для сетки 3×3 .

подтверждение или опровержение существования сдвига пикселя, а также численный вклад в само значение смещения. В связи с тем, что каждое следующее изображение получается прореживанием предыдущего со значением два, значение вектора поля скоростей, соответствующее одному пикселю на l уровне, равняется удвоенным значениям векторов, соответствующих четырём точкам на уровне $l - 1$. В таком случае, для учета каждого вычисления в пирамиде необходимо рассчитывать результирующий вектор поля скоростей по всем представленным уровням, принимая во внимание, что необходимо увеличивать вектор во столько раз, во сколько он отличается по размеру от исходного изображения.

Таким образом, следуя всем вышеперечисленным пунктам, была написана программа, которая рассчитывает поле скоростей для входного набора изображений, состоящего из двух последовательных снимков.

2.6 Тестовые задачи

Как критерий правильности работы реализованного метода используется система дифференциальных уравнений вида:

$$\begin{cases} \dot{x} = ay, \\ \dot{y} = bx. \end{cases} \quad (28)$$

Преимущество данной системы заключается в том, что для различных параметров a, b известна ее интегральная кривая, что позволяет нам впоследствии найти поле скоростей. Используя данное свойство системы, можно проверить верность реализованной программы при заданных параметрах.

При подборе различных параметров a и b , избегая комлектности, возможны несколько видов интегральных кривых:

- при $a > 0, b > 0$ интегральные кривые — гипербола;
- $a < 0, b > 0$ — эллипс;
- $a = -1, b = 1$ — окружность;
- $a = 0$ или $b = 0$ — прямые.

Для каждого случая были выбраны наиболее удобные параметры для расчета поля скоростей. Приведем набор полученных тестовых задач.

Гипербола: При параметрах равных $a = 1, b = 1$ рассматривается семейство гипербол, соответствующих уравнению:

$$x^2 - y^2 = C. \quad (29)$$

Здесь постоянная C может быть выбрана произвольно, вследствие чего можно присвоить ей значение равное 400. График данной функции представлен на рис. 5.

Эллипс: При параметрах равных $a = -\frac{1}{9}, b = \frac{1}{4}$ рассматривается семейство эллипсов, соответствующих уравнению:

$$\frac{x^2}{4} + \frac{y^2}{9} = C. \quad (30)$$

Здесь постоянная C также может быть выбрана произвольно, вследствие чего можно присвоить ей значение равное 400. График данной функции

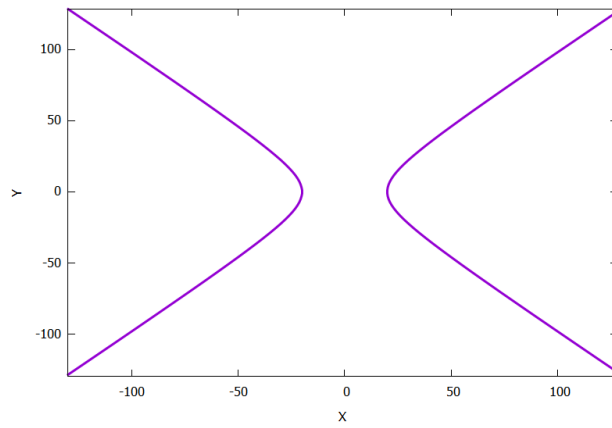


Рис. 5. График функции (29) при $C = 400$.

представлен на рис. 6. Частным случаем эллипса является окружность. Рассмотрим также и его график при $a = -1$, $b = 1$, $C = 1600$ (Рис. 7).

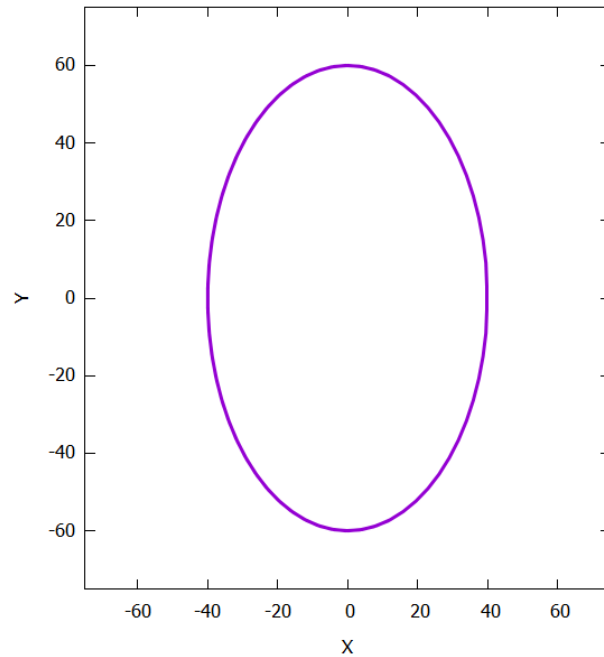


Рис. 6. График функции (30) при $C = 400$.

Прямые: График функций при $a = 0$, $b = 1$, $C = 2500$ приведен на рис. 8.

Из всех вышеприведенных графиков можно заключить, что поле направлений состоит из касательных к их графикам. Для последнего случая поле скоростей идет вдоль самого графика, так как касательная к прямой совпадает с самой прямой.

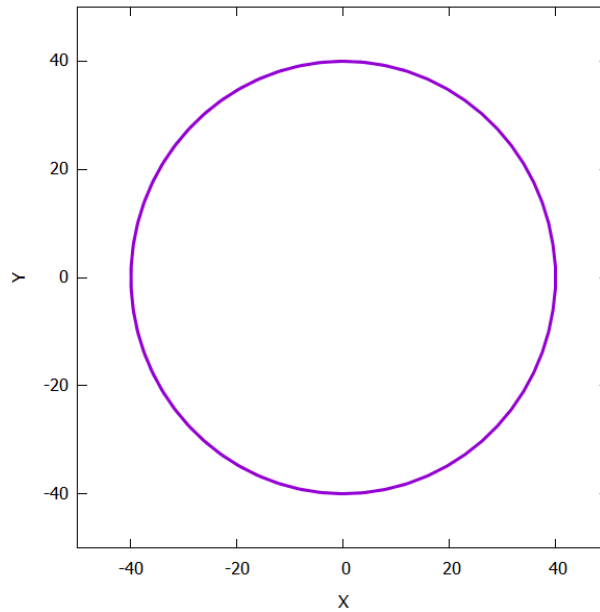


Рис. 7. График функции $x^2 + y^2 = 1600$.

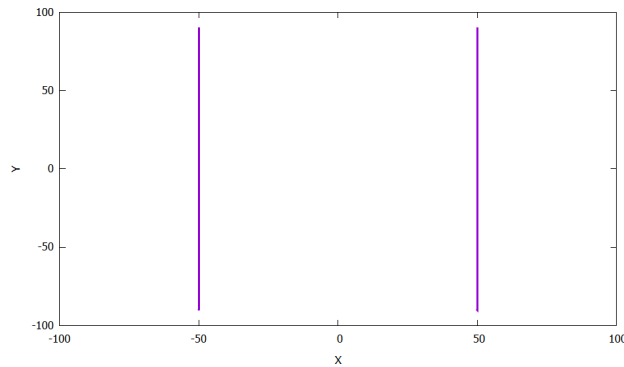


Рис. 8. График функции $x^2 = 2500$.

2.7 Построение входных изображений

Для проверки адекватности работы с помощью системы (28), необходимо было построить представленные в прошлом пункте графики в виде изображений, при этом каждой точке кривой должно соответствовать характерное ей значение интенсивности. Для проведения дальнейших расчетов нужно построить второе изображение, где все точки графика сдвигаются вдоль траектории, а также интенсивность каждой точки меняется по одинаковому закону в соответствии с предположением о постоянстве скорости изменения уровня яркости. В дальнейшем, по представленным двум изображениям будет строиться поле скоростей.

На рис. 9 приведены два изображения со смещением вдоль гиперболы

размером 250×250 . Первое изображение строилось по принципу присвоения каждой точке нечетного значения интенсивности в порядке продвижения по траектории. Для получения второго, каждое значение яркости пикселей увеличивалось на 3, превращаясь в четное значение, и смещалось вдоль траектории на 40 пикселей.

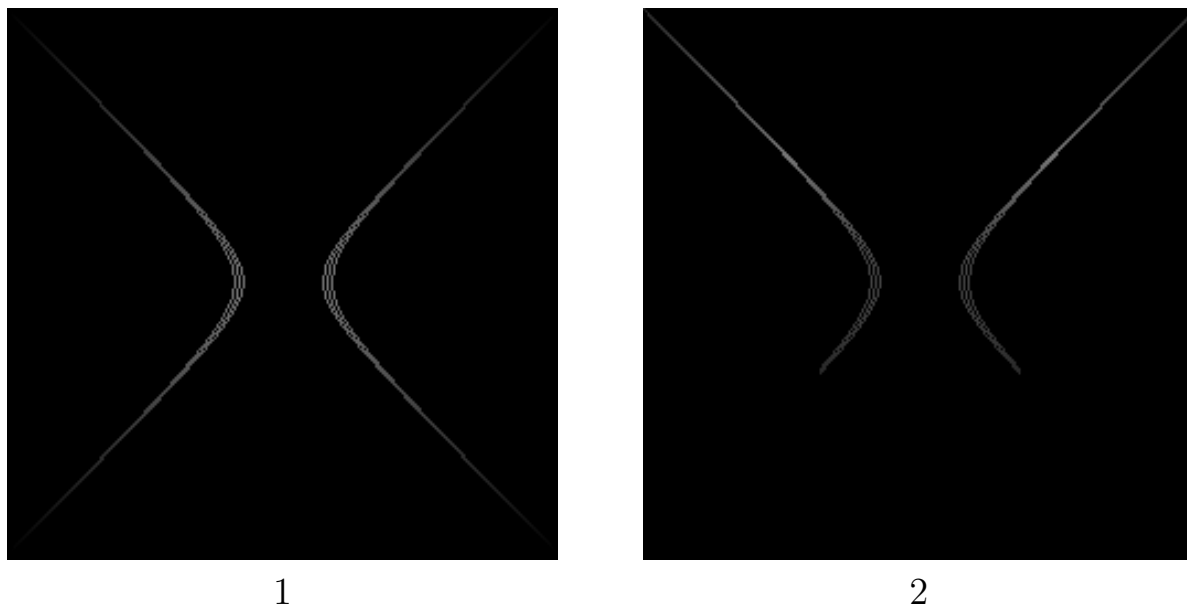
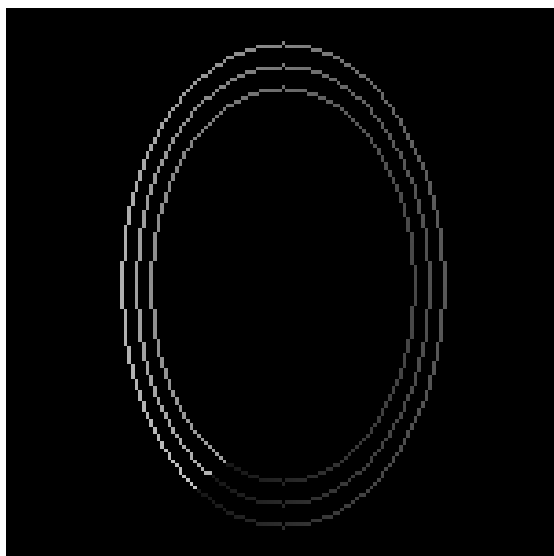


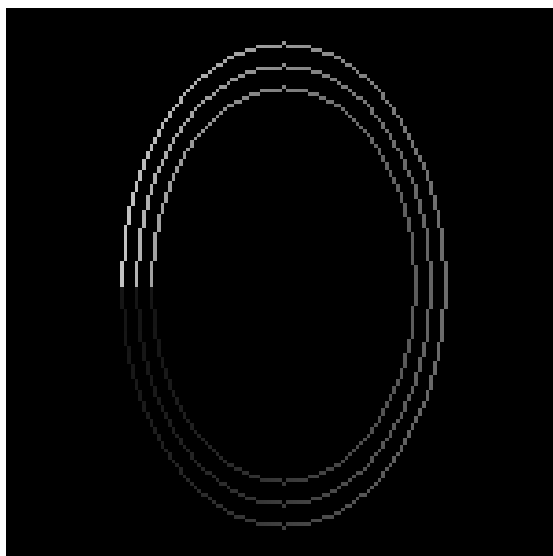
Рис. 9. Смещение точек вдоль гиперболы на 40 пикселей и увеличение интенсивности на 3. 1 и 2 — последовательность изображений размером 250×250 .

На рис. 10 приведены два изображения со смещением вдоль эллипса размером 150×150 . Первое и второе изображения строились аналогично гиперболе. Таким же образом созданы изображения и для сдвига по прямой траектории (рис. 11).

Аналогично, было приведено построение входных изображений для тестовой задачи со смещением вдоль окружности. Здесь особенностью построения является уже не сдвиг вдоль траекторий, а поворот окружности, построенной на первом изображении, на указанный угол с изменением интенсивности на точно такое же число, что и в предыдущих случаях (рис. 12). Для анализа влияния размера смещения на результат работы программы угол вращения выбирается малый. В представленных входных данных угол равен 5° .

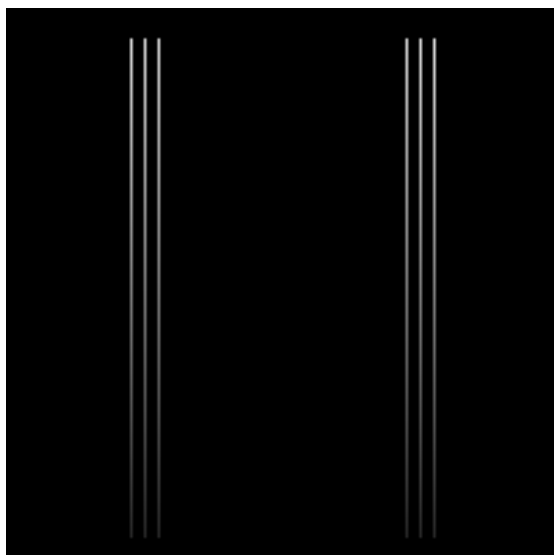


1



2

Рис. 10. Смещение точек вдоль эллипса на 20 пикселей и увеличение интенсивности на 3. 1 и 2 — последовательность изображений размером 150×150 .

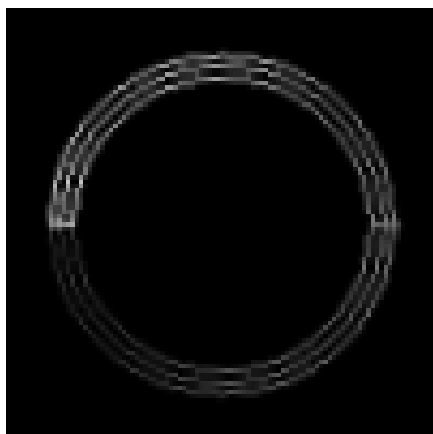


1

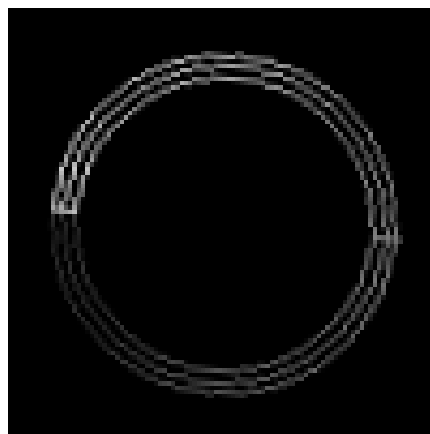


2

Рис. 11. Смещение точек вдоль прямой линии на 40 пикселей и увеличение интенсивности на 3. 1 и 2 — последовательность изображений размером 200×200 .



1



2

Рис. 12. Поворот окружности на угол 5° по часовой стрелке и увеличение интенсивности на 3. 1 и 2 — последовательность изображений размером 100×100 .

2.8 Полученные результаты на тестовых задачах

Результаты, полученные при расчете программы с входными изображениями, представленными выше, и при расчете с помощью системы (28), представлены на рис. 13-16. Все полученные результаты программы совпадают с полем скоростей системы (28) при указанных параметрах a и b .

Как дополнение, приведен результат полученный при обработке изображений представленных на рис. 17 при постоянном значении интенсивности (Рис. 18).

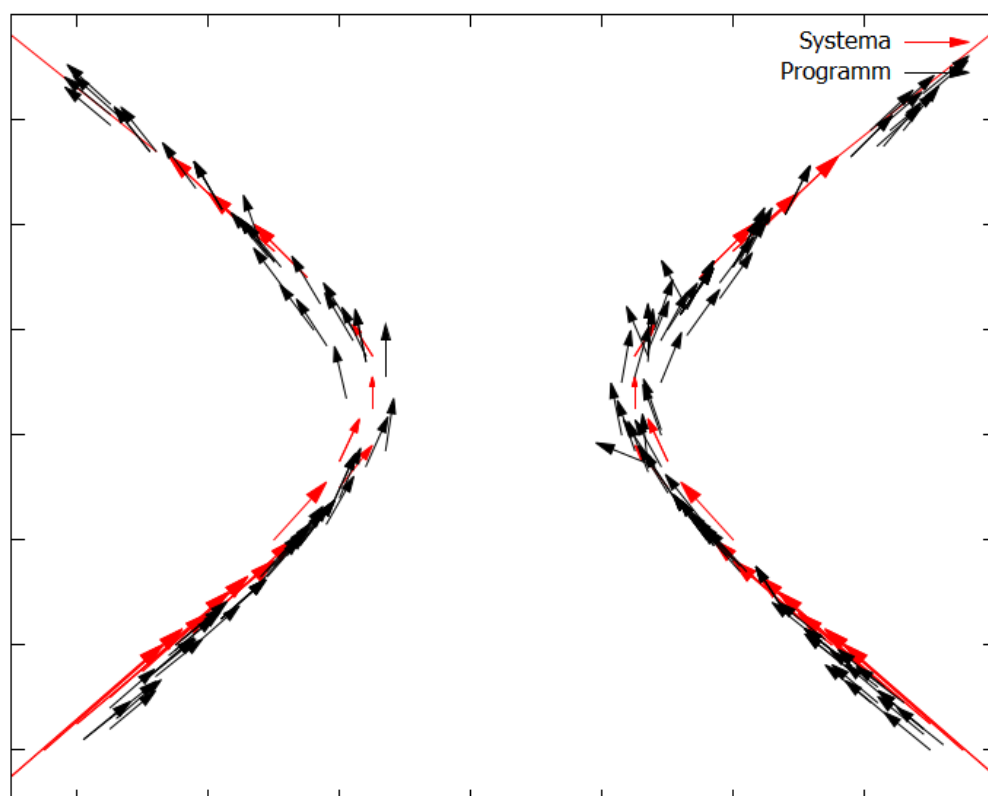


Рис. 13. Результат работы программы с входными изображениями (рис. 9) и расчета с помощью системы (28).

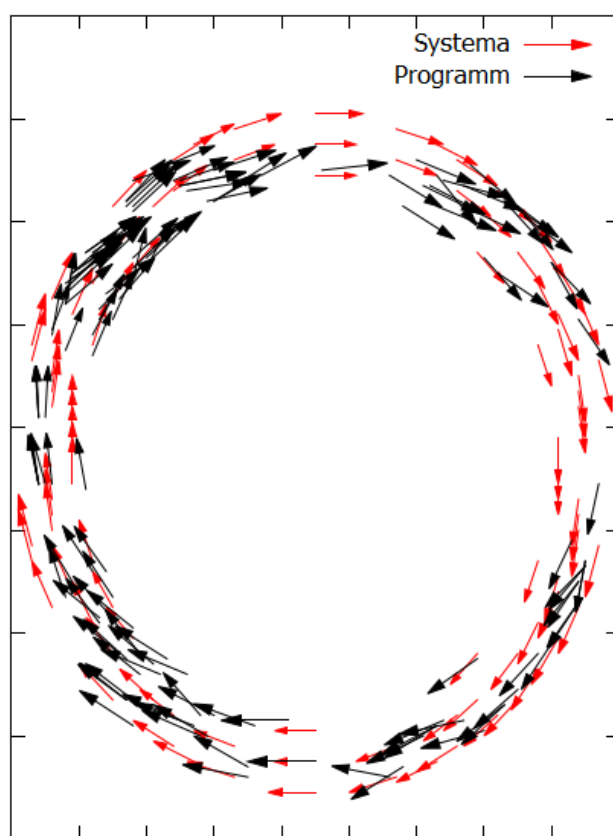


Рис. 14. Результат работы программы с входными изображениями (рис. 10) и расчета с помощью системы (28).

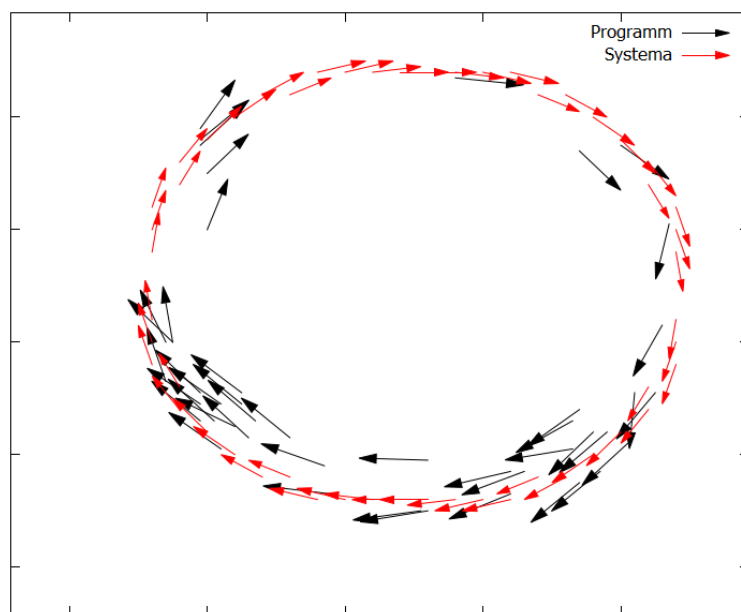


Рис. 15. Результат работы программы с входными изображениями (рис. 12) и расчета с помощью системы (28).

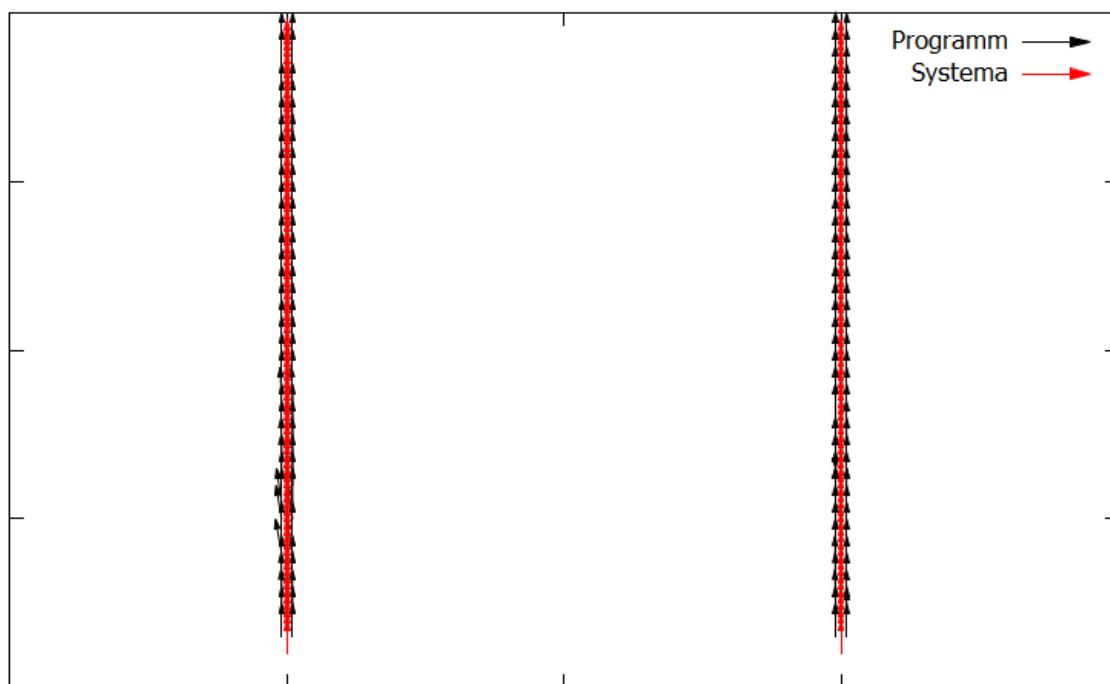
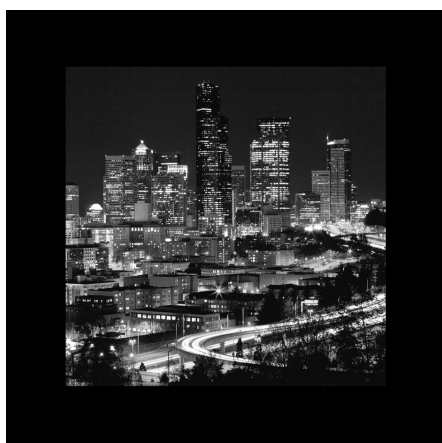
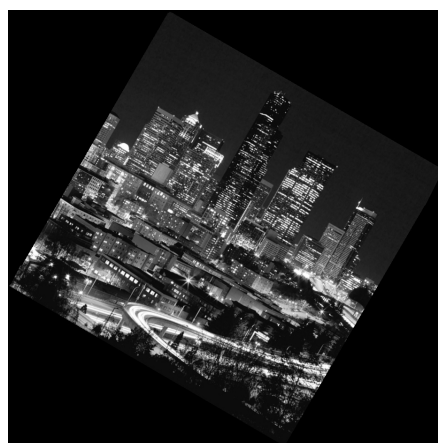


Рис. 16. Результат работы программы с входными изображениями (рис. 11) и расчета с помощью системы (28).



1



2

Рис. 17. Поворот изображения 1 на угол 30° по часовой стрелке. 1 и 2 — последовательность изображений размером 933×933 .

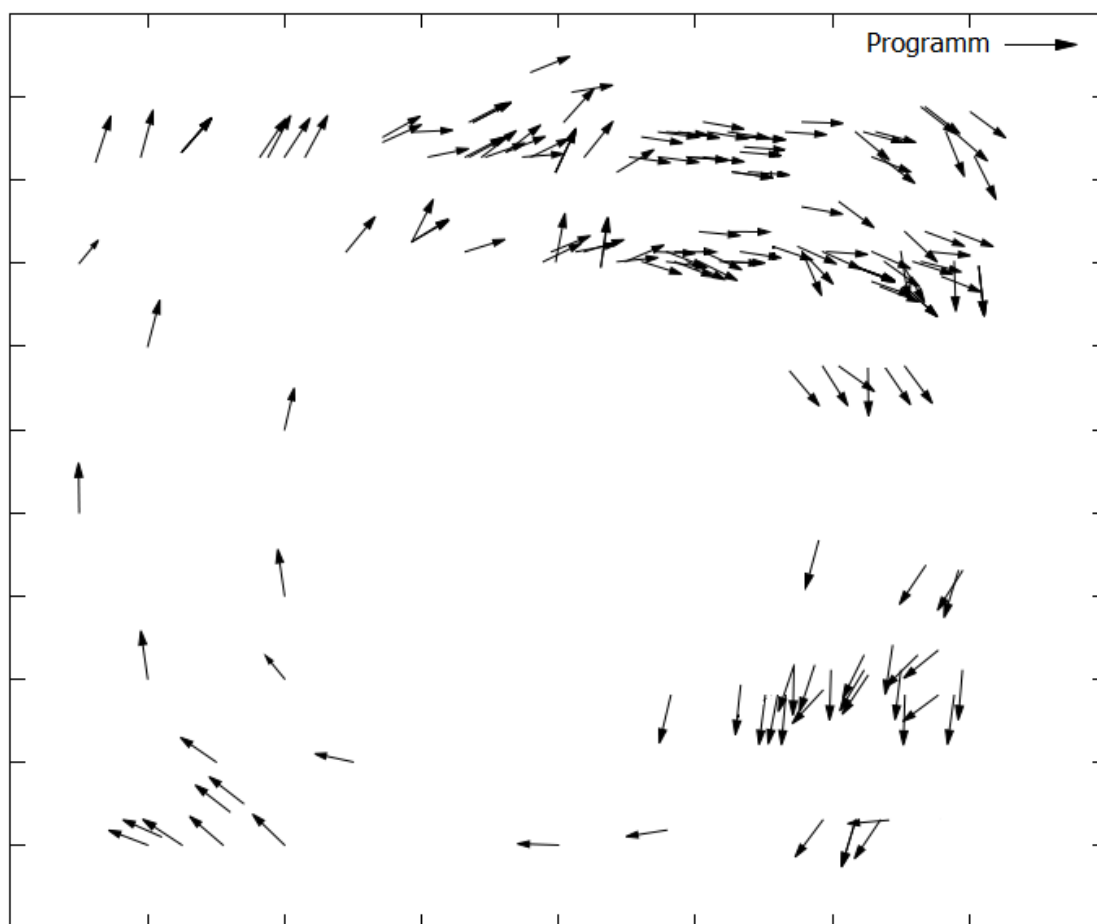


Рис. 18. Результат работы программы с входными изображениями (рис. 17).

Выводы

В настоящее время одним из наиболее актуальных вопросов в построении поля скоростей является задача разработки универсальных алгоритмов и начальных предположений, максимально близких к реальности, для наиболее точного расчета векторов, характеризующих скорость изменения положения предмета или объекта.

В данной работе была описана вычислительная схема, которая не следует традиционным методам оценки оптического потока. В качестве ее базового предположения принято постоянство градиента интенсивности, иными словами скорость изменения яркости точек постоянна. В некоторых случаях, представленный метод дает более точные результаты, чем классический метод оптического потока. Частным случаем может служить пример изменения общего уровня интенсивности изображений.

В работе проведены многочисленные испытания программы с помощью созданных тестовых задач, подтверждающие работоспособность и эффективность реализованного метода. Далее на примере задачи с вращением окружности исследован вопрос о размере смещения, которое может быть обнаружено в рамках рассматриваемого метода.

Заключение

На базе проведенных исследований в ходе выполнения выпускной квалификационной работы получены следующие результаты, которые выносятся на защиту:

- Исследована задача построения поля скоростей методом оптического потока на основе предположения о постоянстве градиента интенсивности, построены алгоритм и расчетная схема;
- Реализован данный метод в среде разработки Microsoft Visual Studio на языке C++ с помощью библиотеки OpenCV для проверки метода и анализа полученных результатов;
- Оттестирована программа реализованного метода на примерах с известным полем скоростей.

Список литературы

- [1] Fleet D.J., Weiss Y. Optical Flow Estimation. Handbook of Mathematical Models in Computer Vision, 2006, 575 p.
- [2] Barron J.L., Fleet D.J., Beauchemin S. Performance of Optical Flow Techniques. IJCV, 1994, P. 43-77.
- [3] Horn B.K.P., Schunck B.G. Determining optical flow. Artificial intelligence, 1981, Vol. 17, P. 185–203.
- [4] Форсайт Д.А. Компьютерное зрение. Современный подход: пер. с англ. Понс Жан. – Издательский дом «Вильямс», 2004, 928с.
- [5] Humphreys G.W., Bruce V. Visual Cognition: Computational, Experimental, and Neuropsychological Perspectives, 1989, 337p.
- [6] Котина Е. Д. . Некоторые вопросы моделирования динамических процессов в радионуклидных исследованиях. ВВМ, 2013, 150с.
- [7] Котина Е. Д., Пасечная Г. А. Определение поля скоростей в задачах обработки изображений, Изв. Иркутского гос. ун-та. Серия: Математика, 2013, том 6, выпуск 3, С. 48–59.
- [8] Котина Е. Д. О сходимости блочных итерационных методов. Серия: Математика, 2012, том 5, выпуск 3, С. 41–55.
- [9] Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. М. : Наука, 1974, 285 с.
- [10] Хорн Б. К. П. Зрение роботов: Пер. с англ. М.: Мир, 1989, 487 с.
- [11] Вазов В., Форсайт Дж. Разностные методы решения дифференциальных уравнений в частных производных. М.: С.-Петербург. ун-та, 1963, 487 с.
- [12] Burt P.J., Adelson E.H. The Laplacian Pyramid as a Compact Image Code. IEE Transaction on Communications, 1983, Vol. 31 No. 4, P. 532-540.